Java - The Map Interface

The Map interface maps unique keys to values. A key is an object that you use to retrieve a value at a later date.

* Given a key and a value, you can store the value in a Map object. After the value is stored, you can retrieve it by using its key.
* Several methods throw a NoSuchElementException when no items exist in the invoking map.
* A ClassCastException is thrown when an object is incompatible with the elements in a map.
* A NullPointerException is thrown if an attempt is made to use a null object and null is not allowed in the map.
* An UnsupportedOperationException is thrown when an attempt is made to change an unmodifiable map.

|  |  |
| --- | --- |
| **Sr.No.** | **Method & Description** |
| 1 | **void clear( )**  Removes all key/value pairs from the invoking map. |
| 2 | **boolean containsKey(Object k)**  Returns true if the invoking map contains **k** as a key. Otherwise, returns false. |
| 3 | **boolean containsValue(Object v)**  Returns true if the map contains **v** as a value. Otherwise, returns false. |
| 4 | **Set entrySet( )**  Returns a Set that contains the entries in the map. The set contains objects of type Map.Entry. This method provides a set-view of the invoking map. |
| 5 | **boolean equals(Object obj)**  Returns true if obj is a Map and contains the same entries. Otherwise, returns false. |
| 6 | **Object get(Object k)**  Returns the value associated with the key **k**. |
| 7 | **int hashCode( )**  Returns the hash code for the invoking map. |
| 8 | **boolean isEmpty( )**  Returns true if the invoking map is empty. Otherwise, returns false. |
| 9 | **Set keySet( )**  Returns a Set that contains the keys in the invoking map. This method provides a set-view of the keys in the invoking map. |
| 10 | **Object put(Object k, Object v)**  Puts an entry in the invoking map, overwriting any previous value associated with the key. The key and value are k and v, respectively. Returns null if the key did not already exist. Otherwise, the previous value linked to the key is returned. |
| 11 | **void putAll(Map m)**  Puts all the entries from **m** into this map. |
| 12 | **Object remove(Object k)**  Removes the entry whose key equals **k**. |
| 13 | **int size( )**  Returns the number of key/value pairs in the map. |
| 14 | **Collection values( )**  Returns a collection containing the values in the map. This method provides a collection-view of the values in the map. |

**General implementations**

**Put(<key>, <Value>)**

|  |  |
| --- | --- |
| **package** com.company; **import** java.util.\*;  **public class** Main {   **public static void** main(String[] args) {  Map m1 = **new** HashMap();  m1.put(**"Zara"**, **"8"**);  m1.put(**"Mahnaz"**, **"31"**);  m1.put(**"Ayan"**, **"12"**);  m1.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements"**);  System.***out***.print(**"\t"** + m1);  } } | Map Elements  {Daisy=14, Ayan=12, Zara=8, Mahnaz=31} |

**.remove(<key> or <key, value>) & .clear()**

|  |  |
| --- | --- |
| **package** com.company; **import** java.util.\*;  **public class** Main {   **public static void** main(String[] args) {  Map m1 = **new** HashMap();  m1.put(**"Zara"**, **"8"**);  m1.put(**"Mahnaz"**, **"31"**);  m1.put(**"Ayan"**, **"12"**);  m1.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements"**);  System.***out***.println(**"\t"** + m1);   System.***out***.println(**"removing elements"**);  *//remove<Key> or <key, value>  //clear* m1.remove(**"Ayan"**);  System.***out***.println(**"\t"** + m1);  *//will no remove anything and not also gonna give any error* System.***out***.println(**"removing elements with unreal value"**);  m1.remove(**"Daisy"**, **"12"**);  System.***out***.println(**"\t"** + m1);  *//real <key, value>* System.***out***.println(**"removing elements with real <key, value> value"**);  m1.remove(**"Daisy"**, **"14"**);  System.***out***.println(**"\t"** + m1);   *//clear* m1.clear();  System.***out***.println(**"removing all elements from the map"**);  System.***out***.println(**"\t"** + m1);  } } | Map Elements  {Daisy=14, Ayan=12, Zara=8, Mahnaz=31}  removing elements  {Daisy=14, Zara=8, Mahnaz=31}  removing elements with unreal value  {Daisy=14, Zara=8, Mahnaz=31}  removing elements with real <key, value> value  {Zara=8, Mahnaz=31}  removing all elements from the map  {} |

**boolean containsKey(Object k) & boolean containsValue(Object v)**

|  |  |
| --- | --- |
| **package** com.company; **import** java.util.\*;  **public class** Main {   **public static void** main(String[] args) {  Map m1 = **new** HashMap();  m1.put(**"Zara"**, **"8"**);  m1.put(**"Mahnaz"**, **"31"**);  m1.put(**"Ayan"**, **"12"**);  m1.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements"**);  System.***out***.println(**"\t"** + m1);   System.***out***.println(**"containsKey(\"<key>\")"**);  **if** (m1.containsKey(**"Zara"**)) *//returns boolean* System.***out***.println(**"key found"**);  **else** System.***out***.println(**"Key not found"**);   System.***out***.println(**"trying - 2 ------------------"**);  **if** (m1.containsKey(**"Zana"**)) *//returns boolean* System.***out***.println(**"key found"**);  **else** System.***out***.println(**"Key not found"**);  *//---------------------------------------------------------------* System.***out***.println(**"----------------------------------------------------"**);  System.***out***.println(**"containsKey(\"<Value>\")"**);  **if** (m1.containsValue(**"14"**)) *//returns boolean* System.***out***.println(**"value found"**);  **else** System.***out***.println(**"value not found"**);   System.***out***.println(**"trying - 2 ------------------"**);  **if** (m1.containsValue(**"18"**)) *//returns boolean* System.***out***.println(**"value found"**);  **else** System.***out***.println(**"value not found"**);  } } | Map Elements  {Daisy=14, Ayan=12, Zara=8, Mahnaz=31}  containsKey("<key>")  key found  trying - 2 ------------------  Key not found  ----------------------------------------------------  containsKey("<Value>")  value found  trying - 2 ------------------  value not found |
| **PART – 2 (get(<key>))** | |
| **package** com.company;**import** java.util.\*;  **public class** Main {   **public static void** main(String[] args) {  Map m1 = **new** HashMap();  m1.put(**"Zara"**, **"8"**);  m1.put(**"Mahnaz"**, **"31"**);  m1.put(**"Ayan"**, **"12"**);  m1.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements"**);  System.***out***.println(**"\t"** + m1);   System.***out***.println(**"containsKey(\"<key>\")"**);  **if** (m1.containsKey(**"Zara"**)) { *//returns boolean* System.***out***.println(**"key found"**);  System.***out***.println(**"the value associated with the key is : "** + m1.values());  System.***out***.println(**"the value associated with the key is (using .get) : "** + m1.get(**"Zara"**));  }  **else** System.***out***.println(**"Key not found"**); } } | Map Elements  {Daisy=14, Ayan=12, Zara=8, Mahnaz=31}  containsKey("<key>")  key found  the value associated with the key is : [14, 12, 8, 31]  the value associated with the key is (using .get) : 8 |
| **package** com.company; *//import javax.xml.bind.SchemaOutputResolver;* **import** java.util.\*;  **public class** Main {   **public static void** main(String[] args) {  Map m1 = **new** HashMap();  m1.put(**"Zara"**, **"8"**);  m1.put(**"Mahnaz"**, **"31"**);  m1.put(**"Ayan"**, **"12"**);  m1.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements"**);  System.***out***.println(**"\t"** + m1);   *//---------------------------------------------------------------* System.***out***.println(**"----------------------------------------------------"**);  System.***out***.println(**"containsKey(\"<Value>\")"**);  **if** (m1.containsValue(**"14"**)) *//returns boolean* System.***out***.println(**"the key associated with the value is : "** + m1.get(**"14"**));  **else** System.***out***.println(**"value not found"**);   } } | Map Elements  {Daisy=14, Ayan=12, Zara=8, Mahnaz=31}  ----------------------------------------------------  containsKey("<Value>")  the key associated with the value is : null  it will return null bcz .get method always accepts a “Key” , not a value |

**Set entryset()**

|  |  |
| --- | --- |
| **package** com.company; *//import javax.xml.bind.SchemaOutputResolver;* **import** java.util.\*;  **public class** Main {   **public static void** main(String[] args) {  Map m1 = **new** HashMap();  m1.put(**"Zara"**, **"8"**);  m1.put(**"Mahnaz"**, **"31"**);  m1.put(**"Ayan"**, **"12"**);  m1.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements"**);  System.***out***.println(**"\t"** + m1);   *//set entryset* System.***out***.println(**"setview of a map"**);  Set x = m1.entrySet();  System.***out***.println(**"\t"** +x);  } } | Map Elements  {Daisy=14, Ayan=12, Zara=8, Mahnaz=31}  setview of a map  [Daisy=14, Ayan=12, Zara=8, Mahnaz=31] |

**Putall(<Map obj>)**

|  |  |
| --- | --- |
| **package** com.company; *//import javax.xml.bind.SchemaOutputResolver;* **import** java.util.\*;  **public class** Main {   **public static void** main(String[] args) {  *//map 1* Map m1 = **new** HashMap();  m1.put(**"Zara"**, **"8"**);  m1.put(**"Mahnaz"**, **"31"**);  m1.put(**"Ayan"**, **"12"**);  m1.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements m2"**);  System.***out***.println(**"\t"** + m1);   *//map 3* Map m3 = **new** HashMap();  m3.put(**"Zoya"**, **"18"**);  m3.put(**"Rick"**, **"30"**);  m3.put(**"Ayan"**, **"12"**);  m3.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements of m3"**);  System.***out***.println(**"\t"** + m3);   m1.putAll(m3);  System.***out***.println(**" putall elements in m1"**);  System.***out***.println(**"\t"** + m1);  } } | Map Elements m2  {Daisy=14, Ayan=12, Zara=8, Mahnaz=31}  Map Elements of m3  {Zoya=18, Rick=30, Daisy=14, Ayan=12}  putall elements in m1  {Zoya=18, Daisy=14, Rick=30, Ayan=12, Zara=8, Mahnaz=31} |

**boolean equals(Object obj)**

|  |  |
| --- | --- |
| **package** com.company; *//import javax.xml.bind.SchemaOutputResolver;* **import** java.util.\*;  **public class** Main {   **public static void** main(String[] args) {  *//map 1* Map m1 = **new** HashMap();  m1.put(**"Zara"**, **"8"**);  m1.put(**"Mahnaz"**, **"31"**);  m1.put(**"Ayan"**, **"12"**);  m1.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements m2"**);  System.***out***.println(**"\t"** + m1);   *//map 2* Map m2 = **new** HashMap();  m2.put(**"Zara"**, **"8"**);  m2.put(**"Mahnaz"**, **"31"**);  m2.put(**"Ayan"**, **"12"**);  m2.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements of m2"**);  System.***out***.println(**"\t"** + m2);   *//map 3* Map m3 = **new** HashMap();  m3.put(**"Zoya"**, **"18"**);  m3.put(**"Rick"**, **"30"**);  m3.put(**"Ayan"**, **"12"**);  m3.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements of m3"**);  System.***out***.println(**"\t"** + m3);   *//CHECKING 1 (m1 => m2)  //set entryset* System.***out***.println();  System.***out***.println(**"checking if objects are equal"**);  **if** (m1.equals(m2))  System.***out***.println(**"Ayan has a age of : "** + m1.get(**"Ayan"**));  **else** System.***out***.println(**"Sorry"**);   *//CHECKING 2 (m1 => m3)* System.***out***.println();  System.***out***.println(**"checking if objects are equal"**);  **if** (m1.equals(m3))  System.***out***.println(**"Ayan has a age of : "** + m1.get(**"Ayan"**));  **else** System.***out***.println(**"Sorry"**);  } } | **Map Elements m2**  **{Daisy=14, Ayan=12, Zara=8, Mahnaz=31}**  **Map Elements of m2**  **{Daisy=14, Ayan=12, Zara=8, Mahnaz=31}**  **Map Elements of m3**  **{Zoya=18, Rick=30, Daisy=14, Ayan=12}**  **checking if objects are equal**  **Ayan has a age of : 12**  **checking if objects are equal**  **Sorry** |

**Set keySet( )**

|  |  |
| --- | --- |
| **package** com.company; *//import javax.xml.bind.SchemaOutputResolver;* **import** java.util.\*;  **public class** Main {   **public static void** main(String[] args) {  *//map 1* Map m1 = **new** HashMap();  m1.put(**"Zara"**, **"8"**);  m1.put(**"Mahnaz"**, **"31"**);  m1.put(**"Ayan"**, **"12"**);  m1.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements m2"**);  System.***out***.println(**"\t"** + m1);   System.***out***.println();  Set v = m1.keySet();  System.***out***.println(**"the keys will be in sets "** + v);  } } | Map Elements m2  {Daisy=14, Ayan=12, Zara=8, Mahnaz=31}  the keys will be in sets [Daisy, Ayan, Zara, Mahnaz] |

**Replace(<Key>, <new val>)**

|  |  |
| --- | --- |
| **package** com.company; *//import javax.xml.bind.SchemaOutputResolver;* **import** java.util.\*;  **public class** Main {   **public static void** main(String[] args) {  *//map 1* Map m1 = **new** HashMap();  m1.put(**"Zara"**, **"8"**);  m1.put(**"Mahnaz"**, **"31"**);  m1.put(**"Ayan"**, **"12"**);  m1.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements m2"**);  System.***out***.println(**"\t"** + m1);   System.***out***.println();  m1.replace(**"Zara"**, **"16"**);  System.***out***.println(**"the keys will be in sets "** + m1);  } } | Map Elements m2  {Daisy=14, Ayan=12, Zara=8, Mahnaz=31}  the keys will be in sets {Daisy=14, Ayan=12, Zara=16, Mahnaz=31} |

**Size()**

|  |  |
| --- | --- |
| **package** com.company; *//import javax.xml.bind.SchemaOutputResolver;* **import** java.util.\*;  **public class** Main {   **public static void** main(String[] args) {  *//map 1* Map m1 = **new** HashMap();  m1.put(**"Zara"**, **"8"**);  m1.put(**"Mahnaz"**, **"31"**);  m1.put(**"Ayan"**, **"12"**);  m1.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements m2"**);  System.***out***.println(**"\t"** + m1);   System.***out***.println();  **int** x = m1.size();  System.***out***.println(**"the keys will be in sets "** + x);  } } | **Map Elements m2**  **{Daisy=14, Ayan=12, Zara=8, Mahnaz=31}**  **the keys will be in sets 4** |

**Collection values( )**

|  |  |
| --- | --- |
| **package** com.company; *//import javax.xml.bind.SchemaOutputResolver;* **import** java.util.\*;  **public class** Main {   **public static void** main(String[] args) {  *//map 1* Map m1 = **new** HashMap();  m1.put(**"Zara"**, **"8"**);  m1.put(**"Mahnaz"**, **"31"**);  m1.put(**"Ayan"**, **"12"**);  m1.put(**"Daisy"**, **"14"**);   System.***out***.println();  System.***out***.println(**" Map Elements m2"**);  System.***out***.println(**"\t"** + m1);   System.***out***.println();  Collection x = m1.values();  System.***out***.println(**"the keys will be in sets "** + x);  } } | Map Elements m2  {Daisy=14, Ayan=12, Zara=8, Mahnaz=31}  the keys will be in sets [14, 12, 8, 31] |